EXHIBIT 4

Taro

Bitcoin Miami 2022

Olaoluwa Osuntokun • CTO @ Lightning Labs

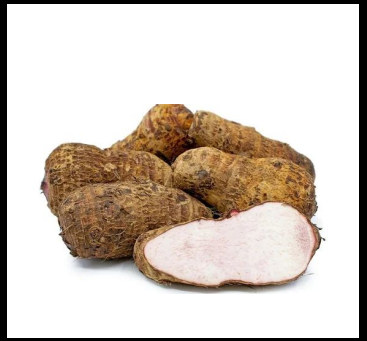
@roasbeef

Outline

- What is Taro?
- How are assets created + structured?
- How do transfers work?
- What is a Universe?
- How does it integrate w/ Lightning Network?

What is Taro?

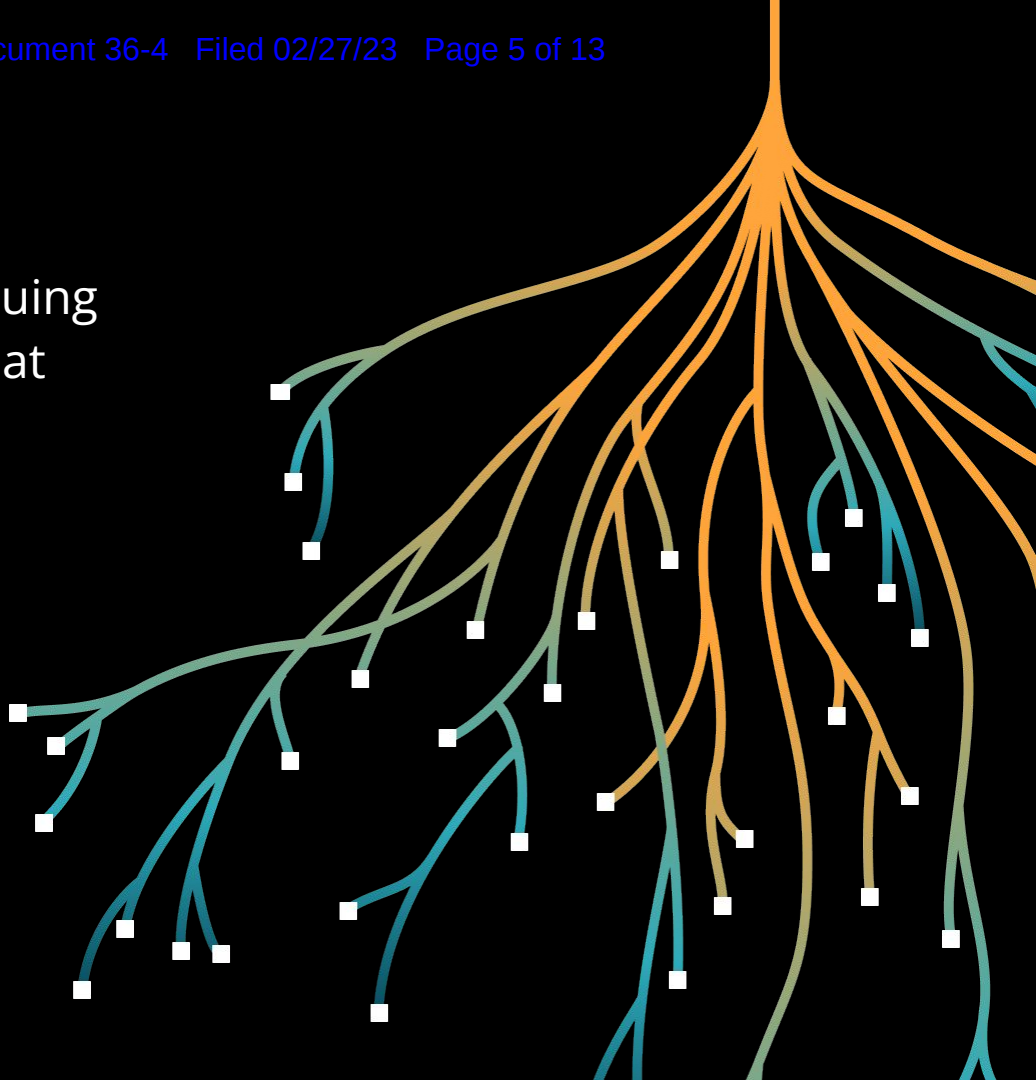
- A type of root vegetable w/ a *taproot* structure
- Eaten across Africa, South America, and Asia
- Great source of manganese, potassium, and fiber
- Nigeria is the largest producer of taro in the world



Taro

A taproot-powered protocol for issuing assets on the Bitcoin blockchain that can be transferred over the Lightning Network.

Taro enables the Lightning Network to move to a multi-asset network with Bitcoin at its core.



What is *Taro*?

Taproot Asset Representation Overlay (Taro)

— What's an asset overlay?

- Based on "embedded consensus":
  - Commit to (include the hash of) data in the main Bitcoin chain (ordering via PoW)
  - Define special "rules" that govern this data (how it can be interacted with)
  - Bitcoin chain used to provide ordering, ensure no double speds

— We use the Taproot script tree to commit to (include the hash of) special asset metadata

- This includes the asset, the asset script hash, the amount, other metadata, etc...
- Support normal divisible assets (beefbux) and collections (1-of-1 holographic beefzard)

Why Taproot?

P2CH techniques have been known for sometime now

- P2CH: key' = `key + h(someData)*G`
- Originally developed as a way to embed a commitment to a receipt within a payment
  - Was mostly pedagogical, never actually deployed to real users

P2SH alone means assets leak into normal Bitcoin Script

- Scripts carry around extra data (hashes or special tweaked keys) not related to the core execution engine

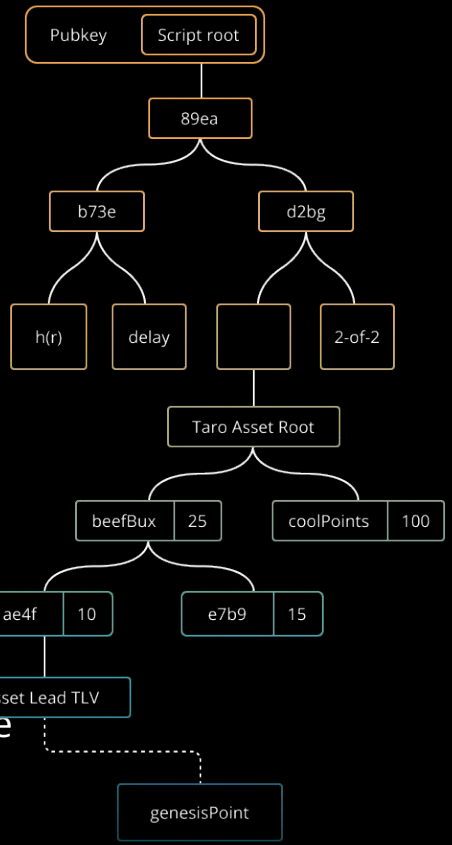
Taproot gives us a standardized top-level commitment

- Able to then layer a series of commitments (YAOLT: layers of indirection)
- Committing to UTXO unlocking + asset unlocking gives us a separation of layers

Taro Asset Trees

Series of nested MS-SMTs rooted in the tapscript tree

- Merkle Sum Sparse Merkle Tree
  - Sum property used for inflation prevention + audits
  - SMT structure for efficient non-inclusion proofs
- Asset ID generation
  - First previous outpoint in asset minting transaction dubbed as the genesisPoint
  - Assumes existence of BIP 34 to guarantee unique asset ID generation
  - `sha256(genesisPoint || assetTag || assetMeta)`



How do transfers work?

Alongside the normal Bitcoin signature/witness, a valid Taro signature/witness is to move assets:

- If rules not abided to, the asset is **burned**
- `asset_version: v1` — Taproot within taproot, using virtual transaction format
- Inherits existing expressibility of Bitcoin Script, able to add new versions in the future (WASM, etc, etc).

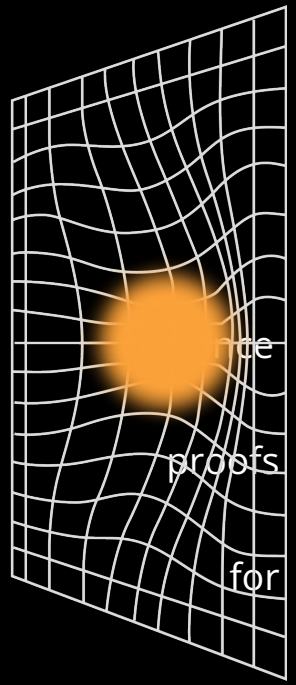
Non-swap transactions (non-interactive) utilize a new address format:

- `bech32(taro, internalKey || assetScriptKey || assetID)`
- Enables light clients support, can't miss-send assets, etc, etc.

Each asset is defined by its *proof:*

- Series of merkle proofs over the *entire history* of the asset
- Rooted at an initial "genesis output"
- Proof size grows linearly with number of transfer

What is a Universe?



All proofs need to be bootstrapped by a canonical starting point:

- A Universe stores the series of genesis outputs for a given asset family
- Stamped on chain in giant special merkle tree that supports proofs”

Each asset can designate a canonical Universe (eg: Rectangle publishing for L-USD)

- Due to “history independence” anyone can maintain a universe a single, or multiple assets (a Multiverse…)

Within a Universe, a “Pocket Universe” can be created to aggregate proofs off-chain

- Scalability mechanism, as single transaction can clear an “unbounded” amount of transactions
- Few different flavors (security model wise), from single sig, to federation, to exit-capable

How does Taro work on LN?

Assets at the Edges:

- LN “core” doesn’t know about assets at all, routers see normal Bitcoin transactions
- Instead, assets exist only at the edges, the “leaf nodes”
- Enables us to retain the existing **network effect**: 3.7k+ BTC already on LN
  - Asset transfer use Bitcoin as the backbone monetary rail
  - more demand for transfers -> more routing activity -> more fee revenue -> network grows and becomes more useful

How does Taro work on LN?

Multi-Hop:

- Taro script v1 is taproot within taproot
- Can re-use scripts for HTLC construct, HTLC exists at two levels: base Bitcoin & Taro
- Asset balances now committed to in the main multi-sig output, as well as settled balances

We're hiring!

https://lightning.engineering/join-us

jobs@lightning.engineering